EE524 Homework 4 : Image Processing with OpenCL

Complete In-class Exercise Ex4 procedures to implement and test the Rotate and GaussBlur kernels using KDF to ensure they work correctly and to determine the optimal local work size configuration.

Create an OpenCL host app which will be used to run the Rotate and GaussBlur kernels

- base this host app on a copy of your performance profiling host app from HW3
- Use stb header tools to read input images and write output (see L6 notes)
- Modify the GaussBlur kernel to accept the sampler and filter as kernel function arguments
- Create and pass a sampler, the 5x5 filter coefficients array (Buffer type), and the filter size as kernel args from the host app

Performance Analysis

- Implement serial 2D convolution-based stencil filtering using C in the host app to use as reference to compare performance against the parallelized OpenCL kernel
- use WPC to time both the serial and OpenCL versions, as in HW3
    - use 100 iterations and compute the mean and stddev execution times
- Compute performance results for several different image sizes
    - use the 5 images provided in HW4 folder on classweb
    - report results for each
    - Produce a plot of image size (# pixels) versus mean execution time for the 5 images
        - can you fit a line to this?
- Calculate the operational intensity (OpI) of GaussBlur kernel (show work).
    - why is OpI for the Rotate kernel problematic?

Modify GaussBlur kernel to use a 7x7 blur stencil (also repeat for 9x9)

- use provided matlab script to generate the filter coefficients
- add new filter array in host app and update host code to pass this filter
    - be sure to pass the right filtersize argument also
- Generate blurred output images of the **Lena_512x512_32bit.png** using
    - original 5x5 stencil
    - 7x7 stencil
        - sigma2 = 0.75 (MATLAB param)
    - 9x9 stencil
        - sigma2 = 1.2
    - put these side-by-side in HW submission for easy comparison